

University of Groningen

## Remark on Algorithm 475. Visible Surface Plotting Program [J6]

Swieten, A.C.M. van; Hosson, J.Th.M. de

*Published in:*  
ACM Transactions on Mathematical Software

*DOI:*  
[10.1145/355853.355870](https://doi.org/10.1145/355853.355870)

**IMPORTANT NOTE:** You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

*Document Version*  
Publisher's PDF, also known as Version of record

*Publication date:*  
1979

[Link to publication in University of Groningen/UMCG research database](#)

### *Citation for published version (APA):*

Swieten, A. C. M. V., & Hosson, J. T. M. D. (1979). Remark on Algorithm 475. Visible Surface Plotting Program [J6]. *ACM Transactions on Mathematical Software*, 5(4), 521-523.  
<https://doi.org/10.1145/355853.355870>

### **Copyright**

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

### **Take-down policy**

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

*Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.*

## REMARK ON ALGORITHM 475

Visible Surface Plotting Program [J6]  
[T. Wright, *Comm. ACM* 17, 3 (March 1974), 152-155]

A.C.M. van Swieten [Recd 28 July 1976 and 12 Sept. 1978]  
VSSG, P.O. Box 3032, Leyden, The Netherlands

and

J.Th.M. de Hosson

Department of Applied Physics, Rijksuniversiteit Groningen, Universiteitscomplex Paddepoel, Nijenborgh 18, 9747 AG Groningen, The Netherlands

This research was supported by the Netherlands Organization for the Advancement of Pure Research (Z.W.O., The Hague).

This remark describes an extension of the visible surface plotting program, ACM Algorithm 475. This program turns out to result in a long plotting time when one is using CALCOMP plot routines. The long plotting time is mainly caused by numerous idle pen movements which are inherent to the structure of the algorithm. Essentially the algorithm does the following: the three-dimensional surface is cut in slices. The slices are separated and then searched in order to produce a perspective image of that slice and to remove the hidden lines; therefore, the algorithm generates a large number of small segments in the search direction. In general, however, the search direction does not coincide with the contour direction. When one is using CALCOMP subroutines there are a lot of idle pen movements due to the fact that the segments are not in an appropriate order. In Figure 1(a) it is shown that numerous idle pen movements are necessary to plot a disklike form. In the improved version only one idle pen movement is made (see Figure 1(b)).

The extension consists of two subroutines: SDLINE and PLTOUT. In the original subroutine DANDR we have to add five statements: Insert

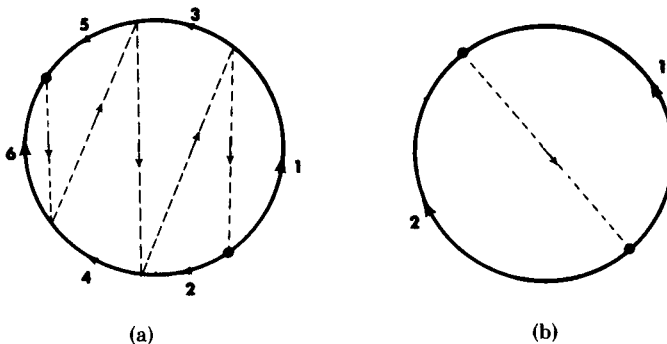


Fig. 1 (a) The pen movements generated by the original version of the plotting program. The idle pen movements are dashed lines. (b) Output of the improved plotting program showing one idle pen movement (dashed line)

COMMON/TOM1/NSEQ, SS; SS = 0.04, NSEQ = 0

before the statement

SLOPE = DX/DY

which initializes

SDLINE; CALL SDLINE(X1, Y1, X2, Y2)

instead of

LINE(X1, Y1, X2, Y2)

which builds up the sequences and

CALL PLTOUT

after the statement

130 CONTINUE

in DANDR which plots the sequences. The subroutine SDLINE(X1, Y1, X2, Y2) temporarily stores the segments in order to construct the sequences. This is done by comparing the last point of each sequence with the endpoints of a segment. The criterion for the continuation of a sequence is that one of the endpoints of the segment lies within a square with edges of  $2SS$  around the last point of a sequence. The value of  $SS$  depends on the plotter precision and it is taken to be equal to 0.04. If there is no continuation point of any sequence a new sequence is started through the segment.

In the present version the length of the sequences is equal to 80 and the number is equal to 20. If a sequence has been filled up completely a new sequence is created. If one needs more than 20 sequences intermediate plotting takes place by calling PLTOUT.

The subroutine PLTOUT plots the sequences taking into account the minimum distance between starting points and ends of sequences. This is done by ordering the sequences in an appropriate way and by indicating whether they should be processed in normal or reversed order.

Finally we give some test results of the revised program compared with the old version. The core size, execution time, and CALCOMP plotting time are compared in the case of the second example (Figure 5) in Algorithm 475. Although this type of surface is not the one that results in the greatest reduction, the saving of plotting time is significant (see Table I). In Table I the time spent in DANDR but not the time spent in PLTOUT is listed. The space of INIT3D + P + DANDR (old version) and of INIT3D + P + DANDR + SDLINE + PLTOUT + TOM are also given in Table I.

Table I

CYBER 74-16	Space	Time	CALCOMP plotting
			time (minutes)
Old version	1277 <sub>s</sub>	5.012	31
Revised version	10544 <sub>s</sub>	5.621	8

**REVISED ALGORITHM**

[Summary information of the revised listing is given here. The complete revised listing is available from the ACM Algorithms Distribution Service (see inside back cover for order form), or may be found in "Collected Algorithms from ACM."]

NAME( $n$ ): indicates a Fortran module with  $n$  records

NAME<sup>T</sup>( $n$ ): indicates "NAME" is included for testing purposes

Contents: CONES<sup>T</sup>(59), INIT3D(122), P(9), DANDR(157), AND(7), OR(7),  
SDLINE(75), PLTOUT(79), ARCCOS(10)

---